

-3-

IN THE SPECIFICATION

Amended claims follow:

1. (Cancelled)

2. (Cancelled)

3. (Currently Amended) ~~The apparatus of claim 1,~~ An apparatus for accelerating a processor running an object-oriented programming language, comprising:

a hardware accelerator interfaced with said processor for implementing at least one application framework of said object-oriented programming language, wherein said at least one application framework comprises a set of classes that embodies an abstract design for solutions to a number of related problems; and

a software stub that controls interfacing of said hardware accelerator with said processor;

wherein said hardware accelerator comprises:

an Input/Output request queue interacting with said processor for receiving and sequentially storing said instructions pending execution of each instruction;

a task processor for processing said instructions from said Input/Output request queue; and

an active object list for tracking the number of reference counts to an instance and for deallocating an instance that is not in use based upon a result of said task processor processing said instructions.

4. (Cancelled)

5. (Currently Amended) ~~The method of claim 4, further comprising the step of~~ A method for accelerating a processor running an object-oriented programming language, comprising the steps of:

implementing at least one application framework of said object-oriented programming language with a hardware accelerator interfaced with said processor, wherein said at least one application framework comprises a set of classes that embodies an abstract design for solutions to a number of related problems;

controlling said interface between said hardware accelerator circuit and said processor with a software stub; and

-4-

managing instances of said application framework and the states of values assigned to said instances using a hardware object management system communicating with said processor and communicating with said hardware accelerator.

6. (Original) The method of claim 5, wherein said step of implementing at least one application framework comprises the steps of:

receiving and sequentially storing said instructions communicated from said processor in an Input/Output request queue pending execution of each instruction; processing said instructions from said Input/Output request queue in a task processor; and

tracking the number of reference counts to an instance and deallocating an instance that is not in use with an active object list based upon interactions with said hardware object management system.

7. (Currently Amended) The method of claim 45, wherein said hardware accelerator is further comprised of a Java.NET circuit to implement the Java.NET and Java.IO application frameworks.

8. (Original) The method of claim 7, wherein the method is further comprised the steps of:
creating and displaying windows and views with a windowing/view system; and
managing a network connection using a connectivity engine in said Java.NET circuit.

9. (Original) The method of claim 5, wherein said hardware object manage system provides object IDs back to a calling system, and wherein said object ID can be of a variable bit length.

10. (Original) The method of claim 5, wherein said hardware object management system can accept object IDs from a calling system to identify, said object identification objects, said method further comprising the steps of:

storing a base object ID in a base reference register and associating said base object ID with a free object location; and

referencing all further object IDs to said base reference register to associate said object ID to an object location.

11. (Currently Amended) The apparatus of claim 1, An apparatus for accelerating a processor running an object-oriented programming language, comprising:

-5-

a hardware accelerator interfaced with said processor for implementing at least one application framework of said object-oriented programming language, wherein said at least one application framework comprises a set of classes that embodies an abstract design for solutions to a number of related problems; and

a software stub that controls interfacing of said hardware accelerator with said processor;
wherein said object-oriented programming language is C++.

12. (Currently Amended) ~~The apparatus of claim 1,~~ An apparatus for accelerating a processor running an object-oriented programming language, comprising:

a hardware accelerator interfaced with said processor for implementing at least one application framework of said object-oriented programming language, wherein said at least one application framework comprises a set of classes that embodies an abstract design for solutions to a number of related problems; and

a software stub that controls interfacing of said hardware accelerator with said processor;
wherein said object-oriented programming language is Java.

13. (Currently Amended) The method of claim 4~~5~~, wherein said object-oriented programming language is C++.

14. (Currently Amended) The method of claim 4~~5~~, wherein said object-oriented programming language is Java.

15. (Cancelled)

16. (Cancelled)

17. (Currently Amended) ~~The apparatus of Claim 15,~~ An apparatus for accelerating a processor running an object-oriented programming language, comprising:

a hardware accelerator interfaces with a said processor for implementing at least one application framework of said object-oriented programming language, wherein said at least one application framework comprises a set of classes that embodies an abstract design for solutions to a number of related problems; and

-6-

a means for a software stub that controls interfacing of said hardware accelerator with said processor;

wherein said hardware accelerator comprises:

an Input/Output request queue interacting with said processor for receiving and sequentially storing said instructions pending execution of each instruction;

a task processor for processing said instructions from said Input/Output request queue; and

a means for an active object list for tracking the number of reference counts to an instance and for deallocating an instance that is not in use based upon a result of said task processor processing said instructions.

18. (Currently Amended) The apparatus of Claim 157, wherein said object-oriented programming language is C++.

19. (Currently Amended) The apparatus of Claim 157, wherein said object-oriented programming language is Java.

20. (Previously Presented) An apparatus for accelerating a processor running the Java programming language, comprising:

a hardware accelerator for implementing at least one Java application framework, wherein said at least one Java application framework comprises a set of classes that embodies an abstract design for solutions to a number of related problems;

a hardware object management system communicatively interfaces with said processor for managing instances of application framework and the states of value assigned to said instances, said hardware object management system comprising

an Input/Output request queue for receiving and sequentially storing said instructions pending execution of each instruction through communication across said communication interface with said processor,

a task processor for processing said instructions received from said Input/Output request queue through interaction with said hardware accelerator, and

a means for an active object list for tracking the number of reference counts to an instance and for deallocating an instance that is not in use base upon a said task processor interaction with said hardware accelerator; and

-7-

a means for software stub of interfacing said hardware object management system with said processor.

21. (Previously Presented) The apparatus of Claim 20, wherein said hardware accelerator further comprising a Java.AWT circuit implementing the Java.AWT application framework through interactions with said hardware object management system.

22. (Previously Presented) The apparatus of Claim 21, wherein said Java.AWT circuit is further comprised of a windowing/view system communicating with said hardware object management system and interacting with a rendering engine.

23. (Previously Presented) The apparatus of Claim 22, wherein said windowing/view system comprise: a general graphics controller communicating with said hardware object management system and communicating with said general graphics controller for managing said frames; a means for a layout manager communicating with said general graphics controller for managing container hierarchies within said container.

24. (Previously Presented) The apparatus of Claim 20, wherein said hardware accelerator further comprising a Java.NET circuit communicating with said hardware object management system implementing the Java.NET and Java.IO application frameworks through communication with said hardware object management system.

25. (Previously Presented) The apparatus of Claim 24, wherein said Java.NET circuit further comprising: a windowing/view system communicating with said task processor and communicating with a connectivity engine.

26. (Previously Presented) The apparatus of Claim 25, wherein said windowing/view system comprises:

a network controller responsive to said hardware object management system, said network controller executing microcode to implement Java.NET framework equivalencies, wherein said network controller operates as an abstraction layer over protocols supported by said connectivity engine through communicating with said connectivity engine;

-8-

a means for network query mechanism for performing DNS lookups and for reporting results to said network controller through communicating with said network controller;

a means for a socket manager for managing sockets in use by applications through communicating with said network controller; and

a means for a stream manager for shuttling input and output data through a link supported by said protocols through communicating with said network controller.

27. (Previously Presented) The apparatus of Claim 20, wherein said hardware accelerator is formed as a part of an application specific integrated circuit.

28. (Previously Presented) An apparatus for accelerating a processor running an object oriented programming language, comprising:

a hardware accelerator interfaced with said processor for implementing at least one application framework of said object-oriented programming language, wherein said at least one application framework comprises a set of classes further comprising:

an Input/Output request queue interacting with said processor for receiving and sequentially storing said instructions pending execution of each instruction;

a task processor for processing said instructions; a task processor for processing said instructions from said Input/Output request queue; and

a means for an active object list for tracking the number of reference counts to an instance and for deallocating an instance that is not in use based upon a result of said task processor processing said instructions;

a hardware object management system interacting with said hardware accelerator for managing instances of said application framework and the states of values assigned to said instances based upon said hardware object management system interacting with said hardware accelerator circuit; and

a means for a software stub that controls interfacing of said hardware accelerator with said processor.

29. (Previously Presented) The apparatus of Claim 28, wherein said hardware accelerator further comprises a Java.AWT circuit interacting with said task processor to implement the Java.AWT circuit interacting with said task processor to implement the Java.AWT application framework through interactions with said task processor.

-9-

30. (Previously Presented) The apparatus of Claim 29, wherein said Java.AWT circuit is further comprised of a windowing/view system communicating with said task processor and interacting with a rendering engine.

31. (Previously Presented) The apparatus of Claim 30, wherein said windowing/view system comprises:

- a general graphics controller communicating with said task processor and interacting with said rendering engine for creating frames and components and for passing data to said rendering engine;

- a window manager communicating with said general graphics controller for managing said frames;

- a layout manager communicating with said general graphics controller for managing component hierarchies within said frame; and

- a component manager communicating with said general graphics controller for managing component hierarchies within said container.

32. (Previously Presented) The apparatus of Claim 28, wherein said hardware accelerator implements the Java.NET and Java.IO application frameworks.

33. (Previously Presented) The apparatus of Claim 32, further comprising: a windowing/view system communicating with said task processor and interacting with a connectivity engine.

34. (Previously Presented) The apparatus of Claim 33, wherein said windowing/view system comprise:

- a network controller responsive to said hardware object management system, said network controller executes microcode to implement Java.NET framework equivalencies, wherein said network controller operates as an abstraction layer over protocols supported by said connectivity engine through interactions between said network controller and said connectivity engine;

- a network query mechanism for performing DSN lookups and for reporting results to said network controller through said network query mechanism interacting with said network controller;

- a socket manager for managing sockets in use by applications through interactions between said socket manager and said network controller; and

-10-

a stream manager for shuttling input and output data through a link supported by said protocols by interactions between said stream manager and said network controller.

35. (Previously Presented) The apparatus of Claim 28, wherein said hardware accelerator is formed as a part of an application specific integrated circuit.

36. (Previously Presented) The apparatus of Claim 28, wherein said object-oriented programming language is Java.

37. (Previously Presented) The apparatus of Claim 28, wherein said object-oriented programming language is C++.